

The equivalence of the torus and the product of two circles in homotopy type theory

Kristina Sojakova

September 2015
CMU-CS-15-105

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Homotopy type theory is a new branch of mathematics which merges insights from abstract homotopy theory and higher category theory with those of logic and type theory. It allows us to represent a variety of mathematical objects as basic type-theoretic constructions, higher inductive types. We present a proof that in homotopy type theory, the torus is equivalent to the product of two circles. This result indicates that the synthetic definition of torus as a higher inductive type is indeed correct.

I gratefully acknowledge the support of the Air Force Office of Scientific Research through MURI grant FA9550-15-1-0053. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the AFOSR.

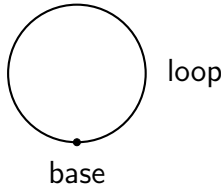
Keywords: homotopy type theory, torus, unit circle, higher inductive type

1 Introduction

Homotopy type theory (HoTT) [15] is a new branch of mathematics which merges insights from abstract homotopy theory and higher category theory with those of logic and type theory. A number of well-known results in algebraic topology have been established within HoTT and formally verified using the proof assistants Agda [11] and Coq [14]; these include the calculation of $\pi_n(\mathbf{S}^n)$ ([9, 7]); the Freudenthal Suspension Theorem [15]; the Blakers-Massey Theorem [15], the van Kampen theorem [15], and the Mayer-Vietoris theorem [1].

As a formal system, HoTT is an extension of Martin-Löf’s dependent type theory with two new concepts: Voevodsky’s *univalence axiom* ([3, 16]) and *higher inductive types* ([10, 12]). The univalence axiom can be paraphrased as stating that *equivalent types are equal*, and hence we can reason about them using the identity elimination principle. While we do not make an explicit use of the axiom in this paper, we use one of its most important consequences - the function extensionality principle - which states that two pointwise equal functions are in fact equal ([2], Ch. 4.9 of [15]).

The second main feature of HoTT, higher inductive types, are a higher-dimensional generalization of ordinary inductive types which allows us to declare constructors involving the *path spaces* of the type X being defined, rather than just X itself. This means that we can define the higher inductive type X *e.g.*, by means of the constructors $\mathbf{base} : X$, $\mathbf{loop} : \mathbf{base} =_X \mathbf{base}$. While \mathbf{base} is an ordinary nullary constructor, akin to the constant 0 in the definition of natural numbers, \mathbf{loop} is a term of an identity type over X , not X itself. Intuitively, we can draw the type X as consisting of the point \mathbf{base} and a loop from \mathbf{base} to \mathbf{base} - also known as the circle:



This is not an isolated occurrence: higher inductive types turn out to be well suited for representing a wide variety of mathematical objects, and the definitions generally require very little prior development. Most of the difficult work then lies in showing that such a “synthetic” definition is indeed the “right” one, in the sense that the higher inductive type representing, *e.g.*, the circle or the torus does possess the expected mathematical properties. For instance, we would like to be able to show that in HoTT, the fundamental group of the circle is the group of integers, and that the torus is the product of two circles.

The former result was shown by [9] and notably, the proof they give is much more concise than its homotopy-theoretic counterpart. In this paper, we present the full proof of the latter result that the torus T^2 is equivalent, in a precise sense, to the product $\mathbf{S}^1 \times \mathbf{S}^1$ of two circles. This problem was brought to the author’s attention during the Special Year on Univalent Foundations at the Institute for Advanced Study in 2012/2013. During that time, the author

gave a sketch of the proof¹ and a year later expanded it into a full writeup [13], which was included in the HoTT Book exercise solutions file but never published. In summer of 2014, Dan Licata and Guillaume Brunerie produced a similar, formalized proof of the result which builds upon their cubical library for the Agda proof assistant. This proof later appeared in a published paper [8]. In the conclusion we provide a more detailed comparison of how the proof presented here compares to the one by Licata and Brunerie.

In [6], Licata later presented a proof of the same result in cubical type theory. This proof is much simpler since the cubical type theory seems better suited for arguments involving higher paths; however, this new theory is itself still under development.

2 Preliminaries

Summarizing from [15], HoTT is a dependent type theory with

- dependent pair types $\Sigma_{x:A} B(x)$ and dependent function types $\Pi_{x:A} B(x)$. The non-dependent versions are denoted by $A \times B$ and $A \rightarrow B$.
- intensional identity types $x =_A y$. We have the usual formation and introduction rules, where the identity path on $x : A$ will be denoted by 1_x . The elimination and computation rules are recalled below:

$$\frac{E : \Pi_{x,y:A} x =_A y \rightarrow \mathbf{type} \quad d : \Pi_{x:A} E(x, x, 1_x)}{J(E, d) : \Pi_{x,y:A} \Pi_{p:x=_A y} E(x, y, p)}$$

$$\frac{E : \Pi_{x,y:A} x =_A y \rightarrow \mathbf{type} \quad d : \Pi_{x:A} E(x, x, 1_x) \quad a : A}{J(E, d)(a, a, 1_a) \equiv d(a) : E(a, a, 1_a)}$$

As usual, these rules are applicable in any context Γ , which we generally omit. If the type $x =_A y$ is inhabited, we call x and y *equal*. If we do not care about the specific equality witness, we often simply say that $x =_A y$. A term $p : x =_A y$ will be often called a *path* and the process of applying the identity elimination rule will be referred to as *path induction*. Definitional equality between $x, y : A$ will be denoted as $x \equiv y : A$.

Proofs of identity behave much like paths in topological spaces: they can be reversed, concatenated, mapped along functions, etc. Below we summarize a few of these properties:

- For any path $p : x =_A y$ there is a path $p^{-1} : y =_A x$, and we have $(1_x)^{-1} \equiv 1_x$.
- For any paths $p : x =_A y$ and $q : y =_A z$ there is a path $p \cdot q : x =_A z$, and we have $1_x \cdot 1_x \equiv 1_x$.
- Associativity of composition: for any paths $p : x =_A y$, $q : y =_A z$, $r : z =_A u$ we have $(p \cdot q) \cdot r \equiv p \cdot (q \cdot r)$.

¹In personal correspondence, P. Lumsdaine stated he also had a sketch of a proof, which has not been made public.

- We have $1_x \cdot p = p$ and $p \cdot 1_y = p$ for any $p : x =_A y$.
- For any $p : x =_A y$, $q : y =_A z$ we have $p \cdot p^{-1} = 1_x$, $p^{-1} \cdot p = 1_y$, and $(p^{-1})^{-1} = p$, $(p \cdot q)^{-1} = q^{-1} \cdot p^{-1}$.
- For any $P : A \rightarrow \mathbf{type}$ and $p : x =_A y$ there is a function $\mathbf{trans}^P(p) : P(x) \rightarrow P(y)$ called the *transport*. We furthermore have $\mathbf{trans}^P(1_x) \equiv \lambda_{x:P(x)}.x$.
- We have $\mathbf{trans}^P(p \cdot q) = \mathbf{trans}^P(q) \circ \mathbf{trans}^P(p)$ for any $P : A \rightarrow \mathbf{type}$ and $p : x =_A y$, $q : y =_A z$.
- For any function $f : A \rightarrow B$ and path $p : x =_A y$, there is a path $\mathbf{ap}_f(p) : f(x) =_B f(y)$ and we have $\mathbf{ap}_f(1_x) \equiv 1_{f(x)}$.
- We have $\mathbf{ap}_f(p^{-1}) = \mathbf{ap}_f(p)^{-1}$ and $\mathbf{ap}_f(p \cdot q) = \mathbf{ap}_f(p) \cdot \mathbf{ap}_f(q)$ for any $f : A \rightarrow B$ and $p : x =_A y$, $q : y =_A z$.
- Given a dependent function $f : \Pi_{x:A} B(x)$ and path $p : x =_A y$, there is a path $\mathbf{apd}_f(p) : \mathbf{trans}^B(p, f(x)) =_{B(y)} f(y)$ and we have $\mathbf{apd}_f(1_x) \equiv 1_{f(x)}$.
- All constructs respect propositional equality.

Definition 1. For $f, g : \Pi_{x:A} B(x)$, we define the type

$$f \sim g := \Pi_{a:A} (f(a) =_{B(a)} g(a))$$

and call it the type of homotopies between f and g .

Definition 2. For $f, g : X \rightarrow Y$, $p : x =_X y$, $\alpha : f \sim g$, there is a path

$$\mathbf{nat}_\alpha(p) : \mathbf{ap}_f(p) \cdot \alpha(y) = \alpha(x) \cdot \mathbf{ap}_g(p)$$

defined in the obvious way by induction on p and referred to as the naturality of the homotopy α . Pictorially, we have

$$\begin{array}{ccc} f(x) & \xrightarrow{\alpha(x)} & g(x) \\ \mathbf{ap}_f(p) \Big\downarrow & \mathbf{nat}_\alpha(p) & \Big\downarrow \mathbf{ap}_g(p) \\ f(y) & \xrightarrow{\alpha(y)} & g(y) \end{array}$$

A crucial concept in HoTT is that of an equivalence between types.

Definition 3. A map $f : A \rightarrow B$ is called an equivalence if it has both a left and a right inverse:

$$\text{iseq}(f) := (\Sigma_{g:B \rightarrow A}(g \circ f \sim \text{id}_A)) \times (\Sigma_{h:B \rightarrow A}(f \circ h \sim \text{id}_B))$$

We define

$$(A \simeq B) := \Sigma_{f:A \rightarrow B} \text{iseq}(f)$$

and call A and B equivalent if the above type is inhabited.

We call A and B *logically equivalent* if there are exist functions $f : A \rightarrow B$, $g : B \rightarrow A$. In practice, we often show that two types A and B are equivalent by first exhibiting the logical equivalence of A and B and then showing that the functions f and g compose to identity on both sides. In this case we refer to f and g as forming a *quasi-equivalence* and say that f and g are *quasi-inverses* of each other. A pair of quasi-inverses can always be turned into an equivalence.

Many “diagram-like” operations on paths turn out to be equivalences. For instance:

- For any $u : a =_A b$, $v : b =_A d$, $w : a =_A c$, $z : c =_A d$, as in the diagram

$$\begin{array}{ccc} a & \xrightarrow{u} & b \\ w \downarrow & & \downarrow v \\ c & \xrightarrow{z} & d \end{array}$$

we have functions

$$\begin{aligned} \mathcal{I} &: (u \cdot v = w \cdot z) \rightarrow (u^{-1} \cdot w \cdot z = v) \\ \mathcal{I}^{-1} &: (u^{-1} \cdot w \cdot z = v) \rightarrow (u \cdot v = w \cdot z) \end{aligned}$$

defined by path induction on u and z , which form a quasi-equivalence.

Finally, we show how to construct paths in pair and function types. Given two pairs $c, d : A \times B$, we can easily construct a function

$$\text{proj}_{c,d}^{\bar{}} : (c = d) \rightarrow (\pi_1(c) = \pi_1(d)) \times (\pi_2(c) = \pi_2(d)).$$

We can show:

Lemma 4. The map $\text{proj}_{c,d}^{\bar{}}$ is an equivalence for any $c, d : A \times B$.

We will denote the quasi-inverse of $\text{proj}_{c,d}^{\bar{}}$ by $\text{pair}_{c,d}^{\bar{}}$. For brevity we will often omit the subscripts.

Analogously, given two functions $f, g : \Pi_{x:A} B(x)$, we can construct a function

$$\text{hap}_{f,g} : (f = g) \rightarrow (f \sim g)$$

Showing that this map is an equivalence (or even constructing a map in the opposite direction) is much harder, and is in fact among the chief consequences of the univalence axiom:

Lemma 5. *The map $\mathbf{hap}_{f,g}$ is an equivalence for any $f, g : \Pi_{x:A} B(x)$.*

Proof. See Ch. 4.9 of [15]. □

We will denote the quasi-inverse of $\mathbf{hap}_{f,g}$ by $\mathbf{funext}_{f,g}$.

3 The circle \mathbf{S}^1 and the torus T^2

The circle \mathbf{S}^1 is a higher inductive type generated by the constructors

$$\begin{aligned} \mathbf{base} &: \mathbf{S}^1, \\ \mathbf{loop} &: \mathbf{base} = \mathbf{base}. \end{aligned}$$

The recursion principle says that given a type $C : \mathbf{type}$ and terms

$$\begin{aligned} b &: C, \\ l &: b = b \end{aligned}$$

there exists a recursor $f : \mathbf{S}^1 \rightarrow C$ for which $f(\mathbf{base}) \equiv b$ and $\mathbf{ap}_f(\mathbf{loop}) = l$. The induction principle says that given a family $E : \mathbf{S}^1 \rightarrow \mathbf{type}$ and terms

$$\begin{aligned} b &: E(\mathbf{base}), \\ l &: \mathbf{trans}^E(\mathbf{loop}, b) = b \end{aligned}$$

there exists an inductor $f : \Pi_{x:\mathbf{S}^1} E(x)$ for which $f(\mathbf{base}) \equiv b$ and $\mathbf{apd}_f(\mathbf{loop}) = l$.

The torus T^2 is a higher inductive type generated by the constructors

$$\begin{aligned} \mathbf{b} &: T^2, \\ \mathbf{p} &: \mathbf{b} = \mathbf{b}, \\ \mathbf{q} &: \mathbf{b} = \mathbf{b}, \\ \mathbf{t} &: \mathbf{p} \cdot \mathbf{q} = \mathbf{q} \cdot \mathbf{p} \end{aligned}$$

as pictured below:

$$\begin{array}{ccc} & \mathbf{p} & \\ \mathbf{b} & \xrightarrow{\quad} & \mathbf{b} \\ \mathbf{q} \downarrow & \Downarrow \mathbf{t} & \downarrow \mathbf{q} \\ \mathbf{b} & \xrightarrow{\quad} & \mathbf{b} \\ & \mathbf{p} & \end{array}$$

The recursion principle says that given a type $C : \mathbf{type}$ and terms

$$\begin{aligned} b' &: C, \\ p' &: b' = b', \\ q' &: b' = b', \\ t' &: p' \cdot q' = q' \cdot p', \end{aligned}$$

there exists a recursor $f : T^2 \rightarrow C$ for which $f(\mathbf{b}) \equiv b'$ and there exist paths $\beta : \mathbf{ap}_f(\mathbf{p}) = p'$ and $\gamma : \mathbf{ap}_f(\mathbf{q}) = q'$ making the following diagram commute:

$$\begin{array}{ccc} \mathbf{ap}_f(p \cdot q) & \xrightarrow{\text{via } \mathbf{t}} & \mathbf{ap}_f(q \cdot p) \\ \downarrow & & \downarrow \\ \mathbf{ap}_f(p) \cdot \mathbf{ap}_f(q) & & \mathbf{ap}_f(q) \cdot \mathbf{ap}_f(p) \\ \downarrow \text{via } \beta, \gamma & & \downarrow \text{via } \gamma, \beta \\ p' \cdot q' & \xrightarrow{t'} & q' \cdot p' \end{array}$$

Here, each edge represents an equality between its vertices. Unlabeled edges stand for the “obvious” equalities which follow from the basic properties of identity types, such as the path from $\mathbf{ap}_f(p \cdot q)$ to $\mathbf{ap}_f(p) \cdot \mathbf{ap}_f(q)$. Edges labeled with, *e.g.*, “via β, γ ” stand for an application of congruence: here β is a path from $\mathbf{ap}_f(p)$ to p' and γ is a path from $\mathbf{ap}_f(q)$ to q' . Since path concatenation respects equality, combining β and γ in a straightforward fashion yields a path from $\mathbf{ap}_f(p) \cdot \mathbf{ap}_f(q)$ to $p' \cdot q'$.

We note that there may be several natural ways how to implement, *e.g.*, the congruence of path concatenation with respect to path equality: we can perform path induction on the first argument, on the second, or on both. For our purposes the exact definition is immaterial as they are all equal up to a higher path, which is why we only specify the arguments (in this case β and γ). From now on, all paths and diagrams will be annotated in this style.

The induction principle for \mathcal{T}^2 is more complicated; it says that given a family $E : T^2 \rightarrow \mathbf{type}$, in order to get an inductor $f : \Pi_{x:T^2} E(x)$ we require terms

$$\begin{aligned} b' &: E(\mathbf{b}) \\ p' &: \mathbf{trans}^E(\mathbf{p}, b') = b' \\ q' &: \mathbf{trans}^E(\mathbf{q}, b') = b' \\ t' &: \left(\mathbf{ap}_{\alpha \mapsto \mathbf{trans}^E(\alpha, b')}(\mathbf{t}) \right)^{-1} \cdot \left(\mathcal{T}_f(E, \mathbf{p}, \mathbf{q}, b') \cdot \mathbf{ap}_{\mathbf{trans}^E(\mathbf{q})}(p') \cdot q' \right) = \\ &\quad \mathcal{T}_f(E, \mathbf{q}, \mathbf{p}, b') \cdot \mathbf{ap}_{\mathbf{trans}^E(\mathbf{p})}(q') \cdot p' \end{aligned}$$

where for any family $E : T^2 \rightarrow \mathbf{type}$, paths $\alpha : x =_{T^2} y$, $\alpha' : y =_{T^2} z$ and point $u : E(x)$, the path

$$\mathcal{T}_f(E, \alpha, \alpha', u) : \mathbf{trans}^E(\alpha \cdot \alpha', u) = \mathbf{trans}^E(\alpha', \mathbf{trans}^E(\alpha, u))$$

is obtained by path induction on α and α' . The inductor f then has the property that $f(b) \equiv b'$. Furthermore, there exist paths $\beta : \mathbf{ap}_f(\mathbf{p}) = p'$ and $\gamma : \mathbf{ap}_f(\mathbf{q}) = q'$ satisfying a higher coherence law, which we omit since we do not need it.

4 Logical equivalence between $\mathbf{S}^1 \times \mathbf{S}^1$ and T^2

Left-to-right We define a function $f : \mathbf{S}^1 \rightarrow T^2$ by circle recursion, mapping $\mathbf{base} \mapsto \mathbf{b}$ and $\mathbf{loop} \mapsto \mathbf{p}$. Thus, we have a definitional equality $f(\mathbf{base}) \equiv \mathbf{b}$ and a path $\beta_f : \mathbf{ap}_f(\mathbf{loop}) = \mathbf{p}$.

We define a function $F^\rightarrow : \mathbf{S}^1 \rightarrow \mathbf{S}^1 \rightarrow T^2$ again by circle recursion, mapping $\mathbf{base} \mapsto f$ and $\mathbf{loop} \mapsto \mathbf{funext}(H)$, where $H : \Pi_{x:\mathbf{S}^1} f(x) = f(x)$ is defined by circle induction as follows. We map \mathbf{base} to \mathbf{q} and \mathbf{loop} to the path

$$\begin{array}{c} \mathbf{trans}^{z \mapsto f(z)=f(z)}(\mathbf{loop}, \mathbf{q}) \\ \left| \mathcal{T}_1(\mathbf{loop}, \mathbf{q}) \right. \\ \mathbf{ap}_f(\mathbf{loop})^{-1} \cdot \mathbf{q} \cdot \mathbf{ap}_f(\mathbf{loop}) \\ \left| \mathcal{I}(\gamma) \right. \\ \mathbf{q} \end{array}$$

where for any $\alpha : x =_{\mathbf{S}^1} y$ and $u : f(x) = f(x)$, the path

$$\mathcal{T}_1(\alpha, u) : \mathbf{trans}^{z \mapsto f(z)=f(z)}(\alpha, u) = \mathbf{ap}_f(\alpha)^{-1} \cdot u \cdot \mathbf{ap}_f(\alpha)$$

is obtained by a straightforward path induction on α , and γ is the path

$$\begin{array}{c} \mathbf{ap}_f(\mathbf{loop}) \cdot \mathbf{q} \\ \left| \text{via } \beta_f \right. \\ \mathbf{p} \cdot \mathbf{q} \\ \left| \mathbf{t} \right. \\ \mathbf{q} \cdot \mathbf{p} \\ \left| \text{via } \beta_f^{-1} \right. \\ \mathbf{q} \cdot \mathbf{ap}_f(\mathbf{loop}) \end{array}$$

Having defined a function $F^\rightarrow : \mathbf{S}^1 \rightarrow \mathbf{S}^1 \rightarrow T^2$, it is now straightforward to define its curried version $F : \mathbf{S}^1 \times \mathbf{S}^1 \rightarrow T^2$. We note that $F^\rightarrow(\mathbf{base}) \equiv f$, and in particular $F(\mathbf{base}, \mathbf{base}) \equiv \mathbf{b}$.

Furthermore, we have a path $\beta_{F \rightarrow} : \mathbf{ap}_{F \rightarrow}(\text{loop}) = \mathbf{funext}(H)$. Since \mathbf{hap} and \mathbf{funext} form a quasi-equivalence, we have a path

$$\beta_{F \rightarrow}^* : \mathbf{hap}(\mathbf{ap}_{F \rightarrow}(\text{loop})) = H$$

The function H is a homotopy between f and f such that $H(\mathbf{base}) \equiv \mathbf{q}$ and the following diagram commutes:

$$\begin{array}{ccc} \mathbf{ap}_f(\text{loop}) \cdot \mathbf{q} & \xrightarrow{\text{nat}_H(\text{loop})} & \mathbf{q} \cdot \mathbf{ap}_f(\text{loop}) \\ \text{via } \beta_f \Big\downarrow & (1) & \Big\downarrow \text{via } \beta_f \\ \mathbf{p} \cdot \mathbf{q} & \xrightarrow{\mathbf{t}} & \mathbf{q} \cdot \mathbf{p} \end{array}$$

To show this, we note that for any $\alpha : x =_{\mathbf{S}^1} y$, applying \mathcal{I}^{-1} to the path

$$\begin{array}{c} \mathbf{ap}_f(\alpha)^{-1} \cdot H(x) \cdot \mathbf{ap}_f(\alpha) \\ \Big\downarrow \mathcal{T}_1(\alpha, H(x))^{-1} \\ \mathbf{trans}^{z \mapsto f(z)=f(z)}(\alpha, H(x)) \\ \Big\downarrow \mathbf{apd}_H(\alpha) \\ H(y) \end{array}$$

yields $\mathbf{nat}_H(\alpha)$: this follows by a path induction on α and a subsequent generalization and path induction on $H(x)$. The second computation rule for H tells us that

$$\mathbf{apd}_H(\text{loop}) = \mathcal{T}_1(\text{loop}, \mathbf{q}) \cdot \mathcal{I}(\gamma)$$

Thus

$$\mathbf{nat}_H(\text{loop}) = \mathcal{I}^{-1}(\mathcal{T}_1(\text{loop}, \mathbf{q})^{-1} \cdot \mathbf{apd}_H(\text{loop})) = \gamma$$

which proves the commutativity of (1).

Finally, we note that for any $\alpha : x =_{T^2} x'$ and $\alpha' : y =_{T^2} y'$, we have path families

$$\begin{aligned} \mu_x(\alpha') &: \mathbf{ap}_F(\mathbf{pair}^=(1_x, \alpha')) = \mathbf{ap}_{F \rightarrow (x)}(\alpha') \\ \nu_y(\alpha) &: \mathbf{ap}_F(\mathbf{pair}^=(\alpha, 1_y)) = \mathbf{hap}(\mathbf{ap}_{F \rightarrow}(\alpha), y) \end{aligned}$$

defined by path induction on α' and α respectively.

Right-to-left We define a function $G : T^2 \rightarrow \mathbf{S}^1 \times \mathbf{S}^1$ by torus recursion as follows. We map $\mathbf{b} \mapsto (\mathbf{base}, \mathbf{base})$, $\mathbf{p} \mapsto \mathbf{pair}^=(1_{\mathbf{base}}, \mathbf{loop})$, $\mathbf{q} \mapsto \mathbf{pair}^=(\mathbf{loop}, 1_{\mathbf{base}})$, and $\mathbf{t} \mapsto \Phi_{\mathbf{loop}, \mathbf{loop}}$, where for any $\alpha : x =_{\mathbf{S}^1} x'$, $\alpha' : y =_{\mathbf{S}^1} y'$, the path

$$\Phi_{\alpha, \alpha'} : \left(\mathbf{pair}^=(1_x, \alpha') \cdot \mathbf{pair}^=(\alpha, 1_{y'}) \right) = \left(\mathbf{pair}^=(\alpha, 1_y) \cdot \mathbf{pair}^=(1_{x'}, \alpha') \right)$$

is defined by induction on α and α' .

Then we have a definitional equality $G(\mathbf{b}) \equiv (\mathbf{base}, \mathbf{base})$ and paths

$$\begin{aligned} \beta_G^p : \mathbf{ap}_G(\mathbf{p}) &= \mathbf{pair}^=(1_{\mathbf{base}}, \mathbf{loop}) \\ \beta_G^q : \mathbf{ap}_G(\mathbf{q}) &= \mathbf{pair}^=(\mathbf{loop}, 1_{\mathbf{base}}) \end{aligned}$$

which make the following diagram commute:

$$\begin{array}{ccc} \mathbf{ap}_G(\mathbf{p} \cdot \mathbf{q}) & \xrightarrow{\text{via } \mathbf{t}} & \mathbf{ap}_G(\mathbf{q} \cdot \mathbf{p}) \\ \downarrow & & \downarrow \\ \mathbf{ap}_G(\mathbf{p}) \cdot \mathbf{ap}_G(\mathbf{q}) & (2) & \mathbf{ap}_G(\mathbf{q}) \cdot \mathbf{ap}_G(\mathbf{p}) \\ \downarrow \text{via } \beta_G^p, \beta_G^q & & \downarrow \text{via } \beta_G^q, \beta_G^p \\ \mathbf{pair}^=(1, \mathbf{loop}) \cdot \mathbf{pair}^=(\mathbf{loop}, 1) & \xrightarrow{\Phi_{\mathbf{loop}, \mathbf{loop}}} & \mathbf{pair}^=(\mathbf{loop}, 1) \cdot \mathbf{pair}^=(1, \mathbf{loop}) \end{array}$$

5 Equivalence between $\mathbf{S}^1 \times \mathbf{S}^1$ and T^2

Left-to-right We need to show that for any $x, y : \mathbf{S}^1$ we have $G(F(x, y)) = (x, y)$. We do this by circle induction on the first argument. We need a path family $\epsilon : \Pi_{y:\mathbf{S}^1} G(f(y)) = (\mathbf{base}, y)$. The definition of ϵ itself proceeds by circle induction: we map \mathbf{base} to the path $1_{(\mathbf{base}, \mathbf{base})}$ and \mathbf{loop} to the path

$$\begin{aligned} & \mathbf{trans}^{z \mapsto G(f(z)) = (\mathbf{base}, z)}(\mathbf{loop}, 1_{(\mathbf{base}, \mathbf{base})}) \\ & \quad \left| \mathcal{T}_2(\mathbf{loop}, 1_{(\mathbf{base}, \mathbf{base})}) \right. \\ & \mathbf{ap}_G(\mathbf{ap}_f(\mathbf{loop}))^{-1} \cdot 1_{(\mathbf{base}, \mathbf{base})} \cdot \mathbf{pair}^=(1_{\mathbf{base}}, \mathbf{loop}) \\ & \quad \left| \mathcal{I}(\delta) \right. \\ & 1_{(\mathbf{base}, \mathbf{base})} \end{aligned}$$

where for any $\alpha : x =_{\mathbf{S}^1} y$ and $u : G(f(x)) = (\mathbf{base}, x)$, the path

$$\mathcal{T}_2(\alpha, u) : \mathbf{trans}^{z \mapsto G(f(z)) = (\mathbf{base}, z)}(\alpha, u) = \mathbf{ap}_G(\mathbf{ap}_f(\alpha))^{-1} \cdot u \cdot \mathbf{pair}^=(1_{\mathbf{base}}, \alpha)$$

is defined by path induction on α and δ is the path

$$\begin{array}{c}
\mathsf{ap}_G(\mathsf{ap}_f(\mathsf{loop})) \cdot 1_{(\mathsf{base}, \mathsf{base})} \\
| \\
\mathsf{ap}_G(\mathsf{ap}_f(\mathsf{loop})) \\
| \text{ via } \beta_f \\
\mathsf{ap}_G(\mathsf{p}) \\
| \beta_G^p \\
\mathsf{pair}^=(1_{\mathsf{base}}, \mathsf{loop}) \\
| \\
1_{(\mathsf{base}, \mathsf{base})} \cdot \mathsf{pair}^=(1_{\mathsf{base}}, \mathsf{loop})
\end{array}$$

This finishes the definition of ϵ . We now need to prove that

$$\mathsf{trans}^{x \mapsto \Pi(y : \mathbf{S}^1) G(F(x, y)) = (x, y)}(\mathsf{loop}, \epsilon) = \epsilon$$

By function extensionality, it suffices to show that for any $y : \mathbf{S}^1$ we have

$$\mathsf{trans}^{x \mapsto \Pi(z : \mathbf{S}^1) G(F(x, z)) = (x, z)}(\mathsf{loop}, \epsilon) y = \epsilon(y)$$

Straightforward path induction shows that for any $\alpha : \mathsf{base} =_{\mathbf{S}^1} x$, we have

$$\mathsf{trans}^{w \mapsto \Pi(z : \mathbf{S}^1) G(F(w, z)) = (w, z)}(\alpha, \epsilon) y = \mathsf{ap}_G(\mathsf{hap}(\mathsf{ap}_{F \rightarrow}(\alpha), y))^{-1} \cdot \epsilon(y) \cdot \mathsf{pair}^=(\alpha, 1_y)$$

It thus suffices to show that

$$\mathsf{ap}_G(\mathsf{hap}(\mathsf{ap}_{F \rightarrow}(\mathsf{loop}), y)) \cdot \epsilon(y) = \epsilon(y) \cdot \mathsf{pair}^=(\mathsf{loop}, 1_y)$$

After simplifying the left endpoint using $\mathsf{hap}(\beta_{F \rightarrow}^*, y)$ it suffices to show that

$$\mathsf{ap}_G(H(y)) \cdot \epsilon(y) = \epsilon(y) \cdot \mathsf{pair}^=(\mathsf{loop}, 1_y)$$

for any $y : \mathbf{S}^1$. We proceed yet again by circle induction. We map base to the path η below:

$$\begin{array}{c}
\mathsf{ap}_G(\mathsf{q}) \cdot 1_{(\mathsf{base}, \mathsf{base})} \\
| \\
\mathsf{ap}_G(\mathsf{q}) \\
| \beta_G^q \\
\mathsf{pair}^=(\mathsf{loop}, 1_{\mathsf{base}}) \\
| \\
1_{(\mathsf{base}, \mathsf{base})} \cdot \mathsf{pair}^=(\mathsf{loop}, 1_{\mathsf{base}})
\end{array}$$

All that now remains to show is

$$\mathbf{trans}^{y \mapsto \mathbf{ap}_G(H(y)) \cdot \epsilon(y) = \epsilon(y) \cdot \mathbf{pair}^=(\mathbf{loop}, 1_y)}(\mathbf{loop}, \eta) = \eta$$

However, this follows at once from the fact that the circle \mathbf{S}^1 , and hence the product $\mathbf{S}^1 \times \mathbf{S}^1$, is a 1-type (as shown *e.g.*, by Licata and Shulman in [9]): this means that for any two points $x, y : \mathbf{S}^1 \times \mathbf{S}^1$, any two paths $\alpha, \alpha' : x = y$, and any two higher paths $\gamma, \gamma' : \alpha = \alpha'$, we necessarily have $\gamma = \gamma'$.

Right-to-left We need to show that for any $x : T^2$ we have $F(G(x)) = x$. We use torus induction with $\mathbf{b}' := 1_{\mathbf{b}}$. We let p' be the path

$$\begin{array}{c} \mathbf{trans}^{x \mapsto F(G(x))=x}(\mathbf{p}, 1_{\mathbf{b}}) \\ \left| \mathcal{T}_3(\mathbf{p}, 1_{\mathbf{b}}) \right. \\ \mathbf{ap}_F(\mathbf{ap}_G(\mathbf{p}))^{-1} \cdot 1_{\mathbf{b}} \cdot \mathbf{p} \\ \left| \mathcal{I}(\kappa_p) \right. \\ 1_{\mathbf{b}} \end{array}$$

where for any $\alpha : x =_{T^2} y$ and $u : F(G(x)) = x$, the path

$$\mathcal{T}_3(\alpha, u) : \mathbf{trans}^{x \mapsto F(G(x))=x}(\alpha, u) = \mathbf{ap}_F(\mathbf{ap}_G(\alpha))^{-1} \cdot u \cdot \alpha$$

is defined by path induction on α , and κ_p is the path in Fig. 1. Similarly, let q' be the path

$$\begin{array}{c} \mathbf{trans}^{x \mapsto F(G(x))=x}(\mathbf{q}, 1_{\mathbf{b}}) \\ \left| \mathcal{T}_3(\mathbf{q}, 1_{\mathbf{b}}) \right. \\ \mathbf{ap}_F(\mathbf{ap}_G(\mathbf{q}))^{-1} \cdot 1_{\mathbf{b}} \cdot \mathbf{q} \\ \left| \mathcal{I}(\kappa_q) \right. \\ 1_{\mathbf{b}} \end{array}$$

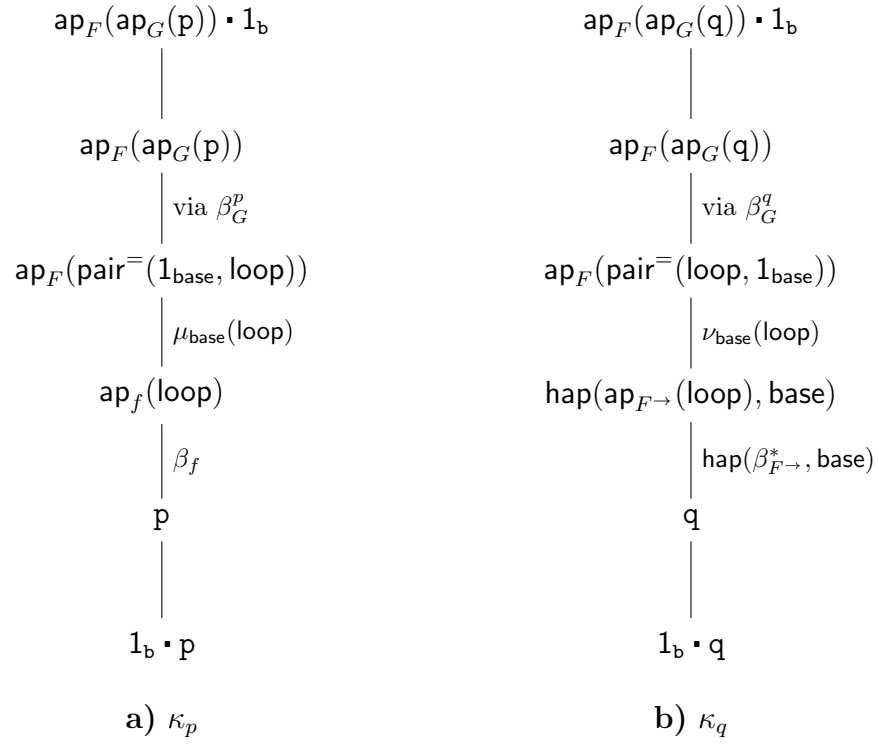


Figure 1: The paths κ_p and κ_q

All that remains now is to show that the following diagram commutes:

$$\begin{array}{ccc}
\text{trans}^{x \mapsto F(G(x))=x}(\mathbf{p} \cdot \mathbf{q}, \mathbf{1}_b) & \xrightarrow{\text{via } \mathbf{t}} & \text{trans}^{x \mapsto F(G(x))=x}(\mathbf{q} \cdot \mathbf{p}, \mathbf{1}_b) \\
\downarrow & & \downarrow \\
\text{trans}^{x \mapsto F(G(x))=x}(\mathbf{q}, \text{trans}^{z \mapsto F(G(x))=x}(\mathbf{p}, \mathbf{1}_b)) & & \text{trans}^{x \mapsto F(G(x))=x}(\mathbf{p}, \text{trans}^{z \mapsto F^\times(G(z))=z}(\mathbf{q}, \mathbf{1}_b)) \\
\downarrow \text{via } p' & & \downarrow \text{via } q' \\
\text{trans}^{x \mapsto F(G(x))=x}(\mathbf{q}, \mathbf{1}_b) & & \text{trans}^{x \mapsto F(G(x))=x}(\mathbf{p}, \mathbf{1}_b) \\
& \searrow q' \quad \swarrow p' & \\
& \mathbf{1}_b &
\end{array}$$

We proceed in four steps.

Step 1 For terms $\alpha_1 : x =_{T^2} y$, $\alpha_2 : y =_{T^2} z$, $\alpha'_1 : a =_{T^2} b$, $\alpha'_2 : b =_{T^2} c$, $u_x : F(G(x)) = a$, $u_y : F(G(y)) = b$, $u_z : F(G(z)) = c$, $\eta_1 : \mathbf{ap}_F(\mathbf{ap}_G(\alpha_1)) \cdot u_y = u_x \cdot \alpha'_1$, $\eta_2 : \mathbf{ap}_F(\mathbf{ap}_G(\alpha_2)) \cdot u_z = u_y \cdot \alpha'_2$, let $\zeta(\alpha_1, \alpha_2, \alpha'_1, \alpha'_2, u_x, u_y, u_z, \eta_1, \eta_2)$ be the path in Fig. 2.

Now for $\alpha_1 : x =_{T^2} y$, $\alpha_2 : y =_{T^2} z$, $u_x : F(G(x)) = x$, $u_y : F(G(y)) = y$, $u_z : F(G(z)) = z$, $\eta_1 : \mathbf{ap}_F(\mathbf{ap}_G(\alpha_1)) \cdot u_y = u_x \cdot \alpha_1$, $\eta_2 : \mathbf{ap}_F(\mathbf{ap}_G(\alpha_2)) \cdot u_z = u_y \cdot \alpha_2$, we claim the path

$$\begin{array}{c}
\text{trans}^{w \mapsto F(G(w))=w}(\alpha_1 \cdot \alpha_2, u_x) \\
\downarrow \\
\text{trans}^{w \mapsto F(G(w))=w}(\alpha_2, \text{trans}^{w \mapsto F(G(w))=w}(\alpha_1, u_x)) \\
\downarrow \text{via } \mathcal{T}_3(\alpha_1, u_x) \cdot \mathcal{I}(\eta_1) \\
\text{trans}^{w \mapsto F(G(w))=w}(\alpha_2, u_y) \\
\downarrow \mathcal{T}_3(\alpha_2, u_y) \cdot \mathcal{I}(\eta_2) \\
u_z
\end{array}$$

is equal to the path

$$\begin{array}{c}
\text{trans}^{w \mapsto F(G(w))=w}(\alpha_1 \cdot \alpha_2, u_x) \\
\downarrow \mathcal{T}_3(\alpha_1 \cdot \alpha_2, u_x) \\
\mathbf{ap}_F(\mathbf{ap}_G(\alpha_1 \cdot \alpha_2))^{-1} \cdot u_x \cdot (\alpha_1 \cdot \alpha_2) \\
\downarrow \text{via } \mathcal{I}(\zeta(\alpha_1, \alpha_2, \alpha_1, \alpha_2, u_x, u_y, u_z, \eta_1, \eta_2)) \\
u_z
\end{array}$$

$$\begin{array}{c}
\mathbf{ap}_F(\mathbf{ap}_G(\alpha_1 \cdot \alpha_2)) \cdot u_z \\
| \\
\left(\mathbf{ap}_F(\mathbf{ap}_G(\alpha_1)) \cdot \mathbf{ap}_F(\mathbf{ap}_G(\alpha_2)) \right) \cdot u_z \\
| \\
\mathbf{ap}_F(\mathbf{ap}_G(\alpha_1)) \cdot \left(\mathbf{ap}_F(\mathbf{ap}_G(\alpha_2)) \cdot u_z \right) \\
| \text{ via } \eta_2 \\
\mathbf{ap}_F(\mathbf{ap}_G(\alpha_1)) \cdot (u_y \cdot \alpha'_2) \\
| \\
\left(\mathbf{ap}_F(\mathbf{ap}_G(\alpha_1)) \cdot u_y \right) \cdot \alpha'_2 \\
| \text{ via } \eta_1 \\
(u_x \cdot \alpha'_1) \cdot \alpha'_2 \\
| \\
u_x \cdot (\alpha'_1 \cdot \alpha'_2)
\end{array}$$

Figure 2: The path $\zeta(\alpha_1, \alpha_2, \alpha'_1, \alpha'_2, u_x, u_y, u_z, \eta_1, \eta_2)$

To show this, we proceed by path induction on α_1 and α_2 . Hence we have to establish the claim for $\alpha_1 := 1_x$, $\alpha_2 := 1_x$, $u_x, u_y, u_z : F(G(x)) = x$, and $\eta_1 : 1_{F(G(x))} \cdot u_y = u_x \cdot 1_x$, $\eta_2 : 1_{F(G(x))} \cdot u_z = u_y \cdot 1_x$.

We note, however, that the types of η_1, η_2 are equivalent to $u_x = u_y$ and $u_y = u_z$ respectively. Hence it suffices to show that given $u_x, u_y, u_z : F(G(x)) = x$, $\eta'_1 : u_x = u_y$, $\eta'_2 : u_y = u_z$, we can establish the claim for the special case when η_1 and η_2 have been obtained from η'_1 and η'_2 , respectively, by using the aforementioned equivalences.

But we can now perform path induction on η'_1 and η'_2 , leaving us with $u_x : F(G(x)) = x$ and $\eta'_1 := 1_{u_x}$, $\eta'_2 := 1_{u_x}$. We finish the proof by generalizing the endpoints of u_x and performing a final path induction.

By what we have just shown, it suffices to prove that the following diagram commutes:

$$\begin{array}{ccc}
\text{trans}^{w \mapsto F(G(w))=w}(\mathbf{p} \cdot \mathbf{q}, 1_{\mathbf{b}}) & \xrightarrow{\text{via } \mathbf{t}} & \text{trans}^{x \mapsto F(G(w))=w}(\mathbf{q} \cdot \mathbf{p}, 1_{\mathbf{b}}) \\
\downarrow \mathcal{T}_3(\mathbf{p} \cdot \mathbf{q}, 1_{\mathbf{b}}) & & \downarrow \mathcal{T}_3(\mathbf{q} \cdot \mathbf{p}, 1_{\mathbf{b}}) \\
\text{ap}_F(\text{ap}_G(\mathbf{p} \cdot \mathbf{q}))^{-1} \cdot 1_{\mathbf{b}} \cdot (\mathbf{p} \cdot \mathbf{q}) & & \text{ap}_F(\text{ap}_G(\mathbf{q} \cdot \mathbf{p}))^{-1} \cdot 1_{\mathbf{b}} \cdot (\mathbf{q} \cdot \mathbf{p}) \\
\searrow \mathcal{I}(\zeta(\mathbf{p}, \mathbf{q}, \mathbf{p}, \mathbf{q}, 1_{\mathbf{b}}, 1_{\mathbf{b}}, 1_{\mathbf{b}}, \kappa_p, \kappa_q)) & & \swarrow \mathcal{I}(\zeta(\mathbf{q}, \mathbf{p}, \mathbf{q}, \mathbf{p}, 1_{\mathbf{b}}, 1_{\mathbf{b}}, 1_{\mathbf{b}}, \kappa_q, \kappa_p)) \\
& 1_{\mathbf{b}} &
\end{array}$$

Step 2 We observe the following: given terms $\alpha, \alpha' : x =_{T^2} y$, $\theta : \alpha = \alpha'$, $u_x : F(G(x)) = x$, $u_y : F(G(y)) = y$, $\eta : \text{ap}_F(\text{ap}_G(\alpha)) \cdot u_y = u_x \cdot \alpha$, and $\eta' : \text{ap}_F(\text{ap}_G(\alpha')) \cdot u_y = u_x \cdot \alpha'$, the commutativity of the diagram

$$\begin{array}{ccc}
\text{trans}^{w \mapsto F(G(w))=w}(\alpha, u_x) & \xrightarrow{\text{via } \theta} & \text{trans}^{w \mapsto F(G(w))=w}(\alpha', u_x) \\
\downarrow \mathcal{T}_3(\alpha, u_x) & & \downarrow \mathcal{T}_3(\alpha', u_x) \\
\text{ap}_F(\text{ap}_G(\alpha))^{-1} \cdot u_x \cdot \alpha & & \text{ap}_F(\text{ap}_G(\alpha'))^{-1} \cdot u_x \cdot \alpha' \\
\searrow \mathcal{I}(\eta) & & \swarrow \mathcal{I}(\eta') \\
& u_y &
\end{array}$$

is equivalent to the commutativity of the diagram

$$\begin{array}{ccc}
\mathsf{ap}_F(\mathsf{ap}_G(\alpha)) \cdot u_y & \xrightarrow{\text{via } \theta} & \mathsf{ap}_F(\mathsf{ap}_G(\alpha')) \cdot u_y \\
\eta \downarrow & & \downarrow \eta' \\
u_x \cdot \alpha & \xrightarrow{\text{via } \theta} & u_x \cdot \alpha'
\end{array}$$

To show this, we proceed by path induction on θ and a subsequent path induction on α . After simplifying it remains to prove that for $u_x, u_y : F(G(x)) = x$, $\eta, \eta' : \mathbf{1}_{F(G(x))} \cdot u_y = u_x \cdot \mathbf{1}_x$, we have $(\mathcal{I}(\eta) = \mathcal{I}(\eta')) \simeq (\eta = \eta')$. But this follows since \mathcal{I} is an equivalence.

By what we have just shown, it suffices to prove that the following diagram commutes:

$$\begin{array}{ccc}
\mathsf{ap}_F(\mathsf{ap}_G(\mathbf{p} \cdot \mathbf{q})) \cdot \mathbf{1}_b & \xrightarrow{\text{via } \mathbf{t}} & \mathsf{ap}_F(\mathsf{ap}_G(\mathbf{q} \cdot \mathbf{p})) \cdot \mathbf{1}_b \\
\zeta(\mathbf{p}, \mathbf{q}, \mathbf{p}, \mathbf{q}, \mathbf{1}_b, \mathbf{1}_b, \mathbf{1}_b, \kappa_p, \kappa_q) \downarrow & & \downarrow \zeta(\mathbf{q}, \mathbf{p}, \mathbf{q}, \mathbf{p}, \mathbf{1}_b, \mathbf{1}_b, \mathbf{1}_b, \kappa_q, \kappa_p) \\
\mathbf{1}_b \cdot (\mathbf{p} \cdot \mathbf{q}) & \xrightarrow{\text{via } \mathbf{t}} & \mathbf{1}_b \cdot (\mathbf{q} \cdot \mathbf{p})
\end{array}$$

Step 3 We observe the following: for $k \in \{1, 2\}$ and $x_1, x_2, x_3 : T^2$ let terms $\alpha_k^1 : x_k = x_{k+1}$; $\alpha_k^2 : G(x_k) = G(x_{k+1})$; $\alpha_k^3, \alpha_k^4 : F(G(x_k)) = F(G(x_{k+1}))$; $\iota_k^1 : \mathsf{ap}_G(\alpha_k^1) = \alpha_k^2$; $\iota_k^2 : \mathsf{ap}_F(\alpha_k^2) = \alpha_k^3$; $\iota_k^3 : \alpha_k^3 = \alpha_k^4$ be given. Then the path

$$\begin{array}{c}
\mathbf{ap}_F(\mathbf{ap}_G(\alpha_1^1 \cdot \alpha_2^1)) \cdot \mathbf{1}_{F(G(x_3))} \\
| \\
\mathbf{ap}_F(\mathbf{ap}_G(\alpha_1^1 \cdot \alpha_2^1)) \\
| \\
\mathbf{ap}_F(\mathbf{ap}_G(\alpha_1^1) \cdot \mathbf{ap}_G(\alpha_2^1)) \\
| \text{ via } \iota_1^1, \iota_2^1 \\
\mathbf{ap}_F(\alpha_1^2 \cdot \alpha_2^2) \\
| \\
\mathbf{ap}_F(\alpha_1^2) \cdot \mathbf{ap}_F(\alpha_2^2) \\
| \text{ via } \iota_1^2, \iota_2^2 \\
\alpha_1^3 \cdot \alpha_2^3 \\
| \text{ via } \iota_1^3, \iota_2^3 \\
\alpha_1^4 \cdot \alpha_2^4 \\
| \\
\mathbf{1}_{F(G(x_1))} \cdot (\alpha_1^4 \cdot \alpha_2^4)
\end{array}$$

is equal to the path $\zeta(\alpha_1^1, \alpha_2^1, \alpha_1^4, \alpha_2^4, \mathbf{1}_{F(G(x_1))}, \mathbf{1}_{F(G(x_2))}, \mathbf{1}_{F(G(x_3))}, \eta_1, \eta_2)$ where η_k is the path

$$\begin{array}{c}
\mathbf{ap}_F(\mathbf{ap}_G(\alpha_k^1)) \cdot \mathbf{1}_{F(G(x_{k+1}))} \\
\downarrow \\
\mathbf{ap}_F(\mathbf{ap}_G(\alpha_k^1)) \\
\downarrow \text{ via } \iota_k^1 \\
\mathbf{ap}_F(\alpha_k^2) \\
\downarrow \iota_k^2 \\
\alpha_k^3 \\
\downarrow \iota_k^3 \\
\alpha_k^4 \\
\downarrow \\
\mathbf{1}_{F(G(x_k))} \cdot \alpha_k^4
\end{array}$$

To show this, we proceed by path induction (with one endpoint fixed) on $\iota_k^1, \iota_k^2, \iota_k^3$ and a subsequent path induction on α_k^1 .

By what we have just shown, it suffices to prove that the outer rectangle in the following diagram commutes:

$$\begin{array}{ccc}
\text{ap}_F(\text{ap}_G(p \cdot q)) \cdot 1 & \xrightarrow{\text{via } \mathfrak{t}} & \text{ap}_F(\text{ap}_G(q \cdot p)) \cdot 1 \\
\downarrow & \text{A} & \downarrow \\
\text{ap}_F(\text{ap}_G(p \cdot q)) & \xrightarrow{\text{via } \mathfrak{t}} & \text{ap}_F(\text{ap}_G(q \cdot p)) \\
\downarrow & & \downarrow \\
\text{ap}_F(\text{ap}_G(p) \cdot \text{ap}_G(q)) & \text{B} & \text{ap}_F(\text{ap}_G(q) \cdot \text{ap}_G(p)) \\
\downarrow \text{via } \beta_G^p, \beta_G^q & & \downarrow \text{via } \beta_G^q, \beta_G^p \\
\text{ap}_F(\text{pair}^=(1, \text{loop}) \cdot \text{pair}^=(\text{loop}, 1)) & \xrightarrow{\text{via } \Phi_{\text{loop}, \text{loop}}} & \text{ap}_F(\text{pair}^=(\text{loop}, 1) \cdot \text{pair}^=(1, \text{loop})) \\
\downarrow & & \downarrow \\
\text{ap}_F(\text{pair}^=(1, \text{loop})) \cdot \text{ap}_F(\text{pair}^=(\text{loop}, 1)) & \text{C} & \text{ap}_F(\text{pair}^=(\text{loop}, 1)) \cdot \text{ap}_F(\text{pair}^=(1, \text{loop})) \\
\downarrow \text{via } \mu_{\text{base}}(\text{loop}), \nu_{\text{base}}(\text{loop}) & & \downarrow \text{via } \nu_{\text{base}}(\text{loop}), \mu_{\text{base}}(\text{loop}) \\
\text{ap}_f(\text{loop}) \cdot \text{hap}(\text{ap}_{F \rightarrow}(\text{loop}), \text{base}) & \xrightarrow{\text{nat}_{\text{hap}(\text{ap}_{F \rightarrow}(\text{loop}))}(\text{loop})} & \text{hap}(\text{ap}_{F \rightarrow}(\text{loop}), \text{base}) \cdot \text{ap}_f(\text{loop}) \\
\downarrow \text{via } \beta_f, \text{hap}(\beta_{F \rightarrow}^*, \text{base}) & \text{D} & \downarrow \text{via } \text{hap}(\beta_{F \rightarrow}^*, \text{base}), \beta_f \\
p \cdot q & \xrightarrow{\mathfrak{t}} & q \cdot p \\
\downarrow & \text{E} & \downarrow \\
1 \cdot (p \cdot q) & \xrightarrow{\text{via } \mathfrak{t}} & 1 \cdot (q \cdot p)
\end{array}$$

Step 4 It suffices to prove that each of the inner rectangles commutes. Rectangles A and E commute obviously. Rectangle B is just diagram (2) transported along ap_F , and hence commutes. Rectangle C commutes by the following generalization: for any $\alpha : x =_{\mathbf{S}^1} y$, the diagram below commutes by path induction on α :

$$\begin{array}{ccc}
\mathsf{ap}_F(\mathsf{pair}^=(1_x, \alpha) \cdot \mathsf{pair}^=(\alpha, 1_y)) & \xrightarrow{\text{via } \Phi_{\alpha, \alpha}} & \mathsf{ap}_F(\mathsf{pair}^=(\alpha, 1_x) \cdot \mathsf{pair}^=(1_y, \alpha)) \\
\downarrow & & \downarrow \\
\mathsf{ap}_F(\mathsf{pair}^=(1_x, \alpha)) \cdot \mathsf{ap}_F(\mathsf{pair}^=(\alpha, 1_y)) & & \mathsf{ap}_F(\mathsf{pair}^=(\alpha, 1_x)) \cdot \mathsf{ap}_F(\mathsf{pair}^=(1_y, \alpha)) \\
\downarrow \text{via } \mu_x(\alpha), \nu_y(\alpha) & & \downarrow \text{via } \nu_x(\alpha), \mu_y(\alpha) \\
\mathsf{ap}_{F \rightarrow (x)}(\alpha) \cdot \mathsf{hap}(\mathsf{ap}_{F \rightarrow}(\alpha), y) & \xrightarrow{\mathsf{nat}_{\mathsf{hap}(\mathsf{ap}_{F \rightarrow}(\alpha))}(\alpha)} & \mathsf{hap}(\mathsf{ap}_{F \rightarrow}(\alpha), x) \cdot \mathsf{ap}_{F \rightarrow (y)}(\alpha)
\end{array}$$

It remains to show that rectangle D commutes. Consider the following diagram:

$$\begin{array}{ccc}
\mathsf{ap}_f(\mathsf{loop}) \cdot \mathsf{hap}(\mathsf{ap}_{F \rightarrow}(\mathsf{loop}), \mathsf{base}) & \xrightarrow{\mathsf{nat}_{\mathsf{hap}(\mathsf{ap}_{F \rightarrow}(\mathsf{loop}))}(\mathsf{loop})} & \mathsf{hap}(\mathsf{ap}_{F \rightarrow}(\mathsf{loop}), \mathsf{base}) \cdot \mathsf{ap}_f(\mathsf{loop}) \\
\downarrow \text{via } \mathsf{hap}(\beta_{F \rightarrow}^*, \mathsf{base}) & D_1 & \downarrow \text{via } \mathsf{hap}(\beta_{F \rightarrow}^*, \mathsf{base}) \\
\mathsf{ap}_f(\mathsf{loop}) \cdot \mathsf{q} & \xrightarrow{\mathsf{nat}_H(\mathsf{loop})} & \mathsf{q} \cdot \mathsf{ap}_f(\mathsf{loop}) \\
\downarrow \text{via } \beta_f & D_2 & \downarrow \text{via } \beta_f \\
\mathsf{p} \cdot \mathsf{q} & \xrightarrow{\mathsf{t}} & \mathsf{q} \cdot \mathsf{p}
\end{array}$$

Commutativity of the outer rectangle clearly implies the commutativity of D. It thus remains to show that D_1 and D_2 commute. The rectangle D_2 is precisely diagram (1), which commutes. Rectangle D_1 commutes by the following generalization: let $\gamma : h_1 =_{f \sim f} h_2$ and $\alpha : x =_{\mathbf{S}^1} y$ be given. Then the following diagram commutes by path induction on γ and α :

$$\begin{array}{ccc}
\mathsf{ap}_f(\alpha) \cdot h_1(y) & \xrightarrow{\mathsf{nat}_{h_1}(\alpha)} & h_1(x) \cdot \mathsf{ap}_f(\alpha) \\
\downarrow \text{via } \mathsf{hap}(\gamma, y) & & \downarrow \text{via } \mathsf{hap}(\gamma, x) \\
\mathsf{ap}_f(\alpha) \cdot h_2(y) & \xrightarrow{\mathsf{nat}_{h_2}(\alpha)} & h_2(x) \cdot \mathsf{ap}_f(\alpha)
\end{array}$$

This finishes the proof.

6 Conclusion

We have presented a homotopy-type theoretic proof that the torus T^2 is equivalent to the product of two circles $\mathbf{S}^1 \times \mathbf{S}^1$. To compare the proof described here to the one given by

Licata and Brunerie in [8], we first note that the definitions of the back-and-forth functions between T^2 and $\mathbf{S}^1 \times \mathbf{S}^1$ are exactly the same. When proving that the functions compose to the identity on $\mathbf{S}^1 \times \mathbf{S}^1$ we used the fact that the circle \mathbf{S}^1 is a 1-type. This simplification is not used by Licata and Brunerie; the lines 75-76, 82-86 in [4] comprise the path algebra which would be avoided by the aforementioned simplification. On the other hand, in this fashion Agda is able to automatically infer the terms `loop1-case` and `loop2-case`, which in our notation correspond to the paths η and δ respectively (of course a paper proof offers no such opportunity).

Similarly, when proving that the functions compose to the identity on \mathcal{T}^2 , the terms `p-case` and `q-case`, which in our proof correspond to the paths κ_q and κ_p , are inferred automatically. Steps 1 and 2 of our proof roughly correspond to lines 403-441 in [5] and 51-57 in [4]; in both proofs, the purpose of these steps is to mediate between a diagram involving transports (a “square-over”) and an equivalent diagram which does not (a “cube”). Steps 3 and 4 then roughly correspond to lines 60-67 in [4]; the commuting diagrams (or “cubes”) established in Step 4 are composed together, using a reordering of operations that is justified by Step 3.

Acknowledgement

The author would like to thank her advisors, Profs. Steve Awodey and Frank Pfenning, for their help.

References

- [1] E. Cavallo. The Mayer-Vietoris sequence in homotopy type theory, 2014. Talk at the Oxford Workshop on Homotopy Type Theory.
- [2] N. Gambino. The univalence axiom and function extensionality, 2011. Talk at the Oberwolfach Mini-Workshop on the Homotopy Interpretation of Constructive Type Theory. Proof due to V. Voevodsky; notes by K. Kapulkin and P. Lumsdaine available at http://www.pitt.edu/~krk56/UA_implies_FE.pdf.
- [3] C. Kapulkin, P. Lumsdaine, and V. Voevodsky. The simplicial model of univalent foundations. Available at arxiv.org as arXiv:1211.2851v1, 2012.
- [4] D. Licata. Agda repository, 2015. Available at <https://github.com/dlicata335/hott-agda/blob/master/homotopy/TS1S1.agda>.
- [5] D. Licata. Agda repository, 2015. Available at <https://github.com/dlicata335/hott-agda/blob/master/lib/cubical/Cube.agda>.
- [6] D. Licata. Cubical type theory, 2015. Talk at the homotopy type theory seminar, Carnegie Mellon University. Slides available from the author’s website.

- [7] D. Licata and G. Brunerie. $\pi_n(\mathbf{S}^n)$ in Homotopy Type Theory. In *Certified Programs and Proofs*, volume 8307 of *LNCs*, pages 1–16. Springer, 2013.
- [8] D. Licata and G. Brunerie. A cubical approach to synthetic homotopy theory. In *Logic in Computer Science (LICS 2015)*, pages 92–103. IEEE, 2015.
- [9] D. Licata and M. Shulman. Calculating the fundamental group of the circle in Homotopy Type Theory. In *Logic in Computer Science (LICS 2013)*, pages 223–232. IEEE Computer Society, 2013.
- [10] P. Lumsdaine. Higher inductive types: a tour of the menagerie, 2011. Post on the Homotopy Type Theory blog.
- [11] U. Norell. *Towards a practical programming language based on dependent type theory*. PhD thesis, Chalmers University of Technology, 2007.
- [12] M. Shulman. Homotopy Type Theory, VI, 2011. Post on the n-category cafe blog.
- [13] K. Sojakova. The torus T^2 is equivalent to the product $\mathbf{S}^1 \times \mathbf{S}^1$ of two circles, 2014. Available at <http://ncatlab.org/homotopytypetheory/files/torus.pdf>.
- [14] Coq Development Team. *The Coq Proof Assistant Reference Manual, version 8.4pl3*. INRIA, 2012. Available at coq.inria.fr.
- [15] The Univalent Foundations Program, Institute for Advanced Study. *Homotopy Type Theory - Univalent Foundations of Mathematics*. Univalent Foundations Project, 2013.
- [16] V. Voevodsky. Univalent foundations of mathematics, 2011. Invited talk at the Workshop on Logic, Language, Information and Computation (WoLLIC 2011).